Instructions

Please upload your project report as a file attachment. The title page should include the title of the project, name of the participants, and a brief abstract. The rest of the report should be typed and double-spaced on standard-sized paper, with 1" margins on all sides. Your report should be approximately 15 pages (with a margin of  pages). If you think it is not possible to fit the material within this page limit, please contact me.

The report should have an introduction with clear description of the problem and (if applicable) references to relevant work on that topic. There should be a methodology section with concise description of the techniques used in the data analysis. This should be followed by the result section where you should discuss the most significant findings (with appropriate tables and plots) of your analysis. If you plan to use multiple algorithms, please create subsections for each of them in the methodology and results sections. The report should conclude with a conclusion section where main takeaways, limitation of the analysis and future direction should be discussed briefly.

Please do not include any R output in the main part of the report. If you want to show results obtained from a R output, please create tables for them to include in the main report. This instruction, however, is not valid for plots. You may put R plots in the main body of the report. Please create a supplementary for the R outputs. The supplementary does not have any page limit. If you want me to look for specific results in the supplementary section, please reference that appropriately in the main report.

A Cluster-Based Approach to Fitting Regression Models

L. Austin Hadamuscin, James Hawkins, Tomiwa Omotesho, Deep Sagar Karki

STAT 6440: Data Mining

Dr. Shuchismita Sarkar

April 23, 2022

Abstract:

Contents

[1. Introduction 4](#_Toc100478224)

[1.1 Motivation 4](#_Toc100478225)

[1.2 Objectives 4](#_Toc100478226)

[2. Data Description 5](#_Toc100478227)

[3. Methodology 6](#_Toc100478228)

[3.1 Leave-One-Out Cross-Validation 6](#_Toc100478229)

[3.2 K-NN 7](#_Toc100478230)

[3.3 K-Means Clustering 7](#_Toc100478231)

[3.3 Model-Based Clustering 8](#_Toc100478232)

[3.4 Random Forest 9](#_Toc100478233)

[3.5 Linear Regression 9](#_Toc100478234)

[4. Results 10](#_Toc100478235)

[5. Appendix 11](#_Toc100478236)

[6. Citations 12](#_Toc100478237)

# 1. Introduction

## 1.1 Motivation

## 1.2 Objectives

# 2. Data Description

# 3. Methodology

## 3.1 Leave-One-Out Cross-Validation

The relatively small number observations in our data allowed us the luxury of using leave-one-out cross validation (LOOCV) whenever we needed to find on optimal parameter. LOOCV uses the following steps:

* partition all data into a training set except for one observation that will be used as our testing set;
* build a model using the training set;
* use this model to predict the response value in the testing set and calculate the mean square error (MSE);
* and repeat this process until every observation has been used in the testing set.

The test MSE is then calculated by taking the average of the MSE*i* values calculated in the LOOCV process. This process is repeated using different parameters and the set of parameters with the lowest test MSE are our optimal parameters. To expedite this process, we used the caret package (Kuhn, 2021) along with other packages in the statistical programming language R (R Core Team, 2021) (RStudio Team, 2021). Using LOOCV over other methods, such as k-fold cross validation, gives us a low bias estimate of test MSE because our training set contains almost all the data; however, because substantial correlation exists between the MSE*i* from the overlapping in the training and testing sets, the variance of the estimate of test MSE is very high, meaning that any addition of new data could result in vastly different outcomes for our optimal parameters. We chose to accept this variance-bias trade-off because each observation is one country partitioning our data into a train, validation, and test set would mean that whole countries would be excluded from the model building process which greatly reduces the applicability of our final model.

## 3.2 k-NN

We used k nearest neighbors (k-NN) to impute missing data. k-NN works by calculating the Euclidean distance between each observation in the data set, and each new observation, : where represents the number of predictors. In classification k-NN, the new value takes the class of the majority among its closest observations based on Euclidean distance. In the case of ties, there are several methods used to classify the new observations such as using the majority class among all observations. Our response variable is continuous, so we used regression k-NN where new observations are assigned the average response of the closest observations. We used this concept to impute our missing data. We data mined the optimal by utilizing the train function from the caret R package (Kuhn, 2021) and the knn method from the FNN R package (Beygelzimer, et al., 2019) to perform LOOCV as described in section 3.1. While doing initial research into k-NN data imputation, we found the function knnImputation from the archived R package DMwR (Torgo, 2010). We modified the code of this function to estimate the missing values for each observation, using the mean of the present values from their nearest neighbors.

## 3.3 K-Means Clustering

We used a -means approach to clustering to cluster our data. The -means clustering algorithm is fairly simple and can be easily explained:

* randomly choose points in -dimentional space to use as the initial centroids;
* calculate the Euclidean distance between the observations and the centroids and assign each observation to its closest centroid;
* recompute new centroids by calculating the center of each group that was created in the previous step. Repeat the previous step;
* The algorithm stops when there is no change to the cluster assignments.

We can measure how well a -means algorithm performs by calculating its within sum of squares () which is a measure of the density of observations in the clusters and is defined as:

where is the set of observations in cluster . The performance of a -means clustering algorithm is highly dependent on the initialization of the centroids created in the first step of the algorithm so to combat this, we randomly generated a set 1000 seeds from a uniform(0,900000) distribution, using the ceiling function (R Core Team, 2021) to round to the next highest whole number. We used theses seeds to run 1000 iterations of the algorithm, choosing the seed that produced the lowest . Using the same 1000 seeds, this process was repeated for and the model that resulted in the lowest without the increase in k resulting in a negligibly lower was chosen.

## 3.3 Model-Based Clustering

We also took a model-based approach to clustering our data. Model-based clustering uses finite mixture models which are a linear combination of weighted density functions:

where are the weights, , , and is the mixture order, or number of components, that use the Expected-Maximization (EM) iterative algorithm to estimate . Like the k-means clustering algorithm, model-based clustering algorithms are highly dependent on initialization so many models must be run to get the optimal assignment of clusters. To determine what set of clusters are the best, a modified version of the likelihood, Bayesian Information Criterion (BIC), is used:

where is the maximum likelihood and is the number of parameters to be estimated. We used the Mclust function from the mclust R package (Scrucca, Fop, Murphy, & Raftery, 2016) to perform the clustering. This package automatically runs the algorithm at different initializations and chooses the best model and returns its negative BIC. We ran the function using K and chose the model with the maximum negative BIC without the increase in K resulting in a negligibly higher negative BIC. We then compared this model, visually, with the model we created using k-means and chose the one that we felt created the best clusters.

## 3.4 Random Forest

## 3.5 Linear Regression

# 4. Results

# 5. Appendix

# 6. Citations

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D., & Li, S. (2019). FNN: Fast Nearest Neighbor Search Algorithms and Applications. *R package version 1.1.3*. Retrieved from https://CRAN.R-project.org/package=FNN

Kuhn, M. (2021). caret: Classification and Regression Training. R package version 6.0-90. Retrieved from https://CRAN.R-project.org/package=caret

R Core Team. (2021). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

RStudio Team. (2021). RStudio: Integrated Development Environment for R. Boston, MA. Retrieved from http://www.rstudio.com/

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal, 8*(1), 289-317. doi:https://doi.org/10.32614/RJ-2016-021

Torgo, L. (2010). Data Mining with R, learning with case studies. Chapman and Hall/CRC. Retrieved from http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR